Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (currently amended) A method for deleting one or more of a plurality of files, the files including one or more chunks stored by a plurality of servers, the method comprising:

identifying a file to be deleted;

renaming the identified file;

permanently deleting the renamed file a predetermined amount of time after renaming the identified file as part of a garbage collection process;

receiving, from the servers, information concerning chunks stored by the servers; and

identifying, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the chunks that ~~do not exist possibly due to the permanent deletion of the renamed file~~ <u>corresponds to the permanently deleted file</u>.

2.   (original) The method of claim 1, wherein the identifying a file to be deleted includes:

receiving a deletion instruction regarding the file.

3.   (original) The method of claim 2, further comprising:

receiving an un-deletion instruction regarding the file; and

restoring an original name to the file without permanently deleting the renamed file.

4.   (original) The method of claim 1, wherein the predetermined amount of time is a

user-configurable amount of time.

5.   (original) The method of claim 4, wherein the user-configurable amount of time

differs for different ones of the files.

6.   (original) The method of claim 1, wherein metadata is associated with the files;

and

wherein the permanently deleting the renamed file includes erasing the metadata

associated with the renamed file.

7.   (currently amended) The method of claim 1, further comprising:

deleting, by the one of the servers, the ~~ones~~ one of the chunks that ~~do not exist~~

corresponds to the permanently deleted file.

8.   (currently amended) The method of claim 1, further comprising:

identifying an orphaned ~~ones-of-the-chunks~~ chunk, including:

        providing a mapping of file names to chunks, and

        identifying a chunk ~~chunks~~, as the orphaned ~~chunks~~ chunk, that ~~are~~ is not

reachable from any of the file names; and

deleting the orphaned ~~chunks~~ chunk.

9.   (currently amended) The method of claim 8, wherein metadata is associated with

the chunks; and

wherein the deleting the orphaned ~~chunks~~ chunk includes erasing the metadata associated with the orphaned ~~chunks~~ chunk.

10. (currently amended) The method of claim 1, further comprising:

maintaining versions of the chunks;

identifying a stale ~~ones of the chunks~~ chunk based on the versions of the chunks; and

deleting the stale ~~chunks~~ chunk.

11. (currently amended) The method of claim 10, wherein metadata is associated with the chunks; and

wherein the deleting the stale ~~chunks~~ chunk includes erasing the metadata associated with the stale ~~chunks~~ chunk.

12. (currently amended) A system for deleting a file that includes data stored by a plurality of servers, comprising:

means for identifying a file to be deleted;

means for logging deletion of the identified file;

means for permanently deleting the file during a garbage collection process that occurs after logging deletion of the identified file;

means for receiving, from the servers, information concerning data stored by the servers; and

means for identifying, to <u>one of</u> the servers, that of the data that ~~does not exist possibly~~ ~~due to the permanent deletion of the renamed file~~ <u>corresponds to the file that was permanently</u> <u>deleted</u>.

13.   (currently amended) A file system, comprising:

a plurality of servers configured to store files as chunks, each of the files including one or more of the chunks; and

a master connected to the servers and configured to:

identify one of the files to be deleted,

rename the identified file,

permanently delete one or more chunks associated with the renamed file a predetermined amount of time after renaming the identified file as part of a garbage collection process,

receive, from the servers, information concerning chunks stored by the servers, and

identify, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the chunks that ~~do not exist, the ones of~~ ~~the chunks including~~ <u>corresponds to one of</u> the one or more permanently deleted chunks.

14.   (currently amended) A method for deleting orphaned chunks of a plurality of chunks stored by a plurality of servers, the method comprising:

providing a mapping of file names to chunks;

identifying chunks, as orphaned chunks, that are not reachable from any of the file names;

deleting the orphaned chunks;

receiving, from the servers, information concerning chunks stored by the servers; and

identifying, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the chunks that ~~are~~ <u>corresponds to one of the deleted</u> orphaned chunks.

15.   (original) The method of claim 14, wherein metadata is associated with the chunks; and

wherein the deleting the orphaned chunks includes erasing the metadata associated with the orphaned chunks.

16.   (currently amended) The method of claim 14, further comprising:

deleting, by <u>the one of</u> the servers, the <u>one of the chunks that corresponds to one of the</u> orphaned chunks.

17.   (original) The method of claim 14, wherein the deletion of the orphaned chunks occurs as part of a garbage collection process.

18.   (currently amended) A system for deleting orphaned chunks of a plurality of chunks stored by a plurality of servers, comprising:

means for mapping file names to chunks;

means for identifying chunks, as orphaned chunks, that are not reachable from any of the file names;

means for deleting the orphaned chunks as part of a garbage collection process;

means for receiving, from the servers, information concerning chunks stored by the servers; and

means for identifying, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the chunks that ~~are~~ <u>corresponds</u> <u>to one of the deleted</u> orphaned chunks.

19.   (currently amended) A file system, comprising:

a plurality of servers configured to store files as chunks, each of the files including one or more of the chunks; and

a master connected to the servers and configured to:

map file names to chunks,

identify chunks, as orphaned chunks, that are not reachable from any of the file names,

delete the orphaned chunks,

receive, from the servers, information concerning chunks stored by the servers, and

identify, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the chunks that ~~are~~ <u>corresponds to one</u> <u>of the deleted</u> orphaned chunks.

20.   (currently amended) A method for deleting stale replicas of chunks, the replicas being stored by a plurality of servers, the method comprising:

associating version information with replicas of chunks;

identifying stale replicas based on the associated version information;

deleting the stale replicas;

receiving, from the servers, information concerning replicas stored by the servers; and

identifying, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the replicas that ~~are~~ <u>corresponds to one of</u>

<u>the deleted</u> stale replicas.

21.   (currently amended) The method of claim 20, wherein the version information for

one of the replicas is updated each time a lease is granted ~~to~~ <u>for</u> the ~~replica~~ <u>one of the replicas.</u>

22.   (currently amended) The method of claim 20, further comprising:

deleting, by <u>the one of</u> the servers, the <u>one of the replicas that corresponds to one of the</u>

stale replicas.

23.   (original) The method of claim 20, wherein the deletion of the stale replicas

occurs as part of a garbage collection process.

24.   (currently amended) A system for deleting stale replicas of chunks, the replicas

being stored by a plurality of servers, the system comprising:

means for generating version information for replicas of chunks;

means for identifying stale replicas based on the generated version information;

means for deleting the stale replicas as part of a garbage collection process;

means for receiving, from the servers, information concerning replicas stored by the

servers; and

means for identifying, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the replicas that ~~are~~ <u>corresponds</u> <u>to one of the deleted</u> stale replicas.

25.   (currently amended) A file system that stores files as chunks, comprising:

a plurality of servers configured to store files as chunks; and

a master connected to the servers and configured to:

        associate version information with the chunks,

        identify stale chunks based on the associated version information,

        delete the stale chunks,

        receive, from the servers, information concerning replicas stored by the servers,

and

        identify, to <u>one of</u> the servers, ~~ones~~ <u>one</u> of the replicas that ~~are stale replicas~~

<u>corresponds to one of the deleted stale chunks</u>.